УДК 519.854.2

ПАВЛОВ А.А.,
СПЕРКАЧ М.О.,
ХАЛУС Е.А.

СУБОПТИМАЛЬНЫЙ ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ОДНОГО КЛАССА МНОГОЭТАПНЫХ СЕТЕВЫХ ЗАДАЧ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ

В статье рассматривается один класс многоэтапной сетевой задачи календарного планирования. Для произвольно заданных конечных директивных сроков (директивных сроков выпуска готовых изделий) необходимо получить допустимое расписание с максимально поздним по времени запуском технологического процесса. При дополнительном ограничении на структуру допустимого расписания излагается точный полиномиальный алгоритм решения сформулированной задачи. Приводится теоретическое и практическое обоснование целесообразности введения дополнительного ограничения на структуру искомого расписания.

In this paper we consider one class of multistage network calendar scheduling problem. For arbitrarily given finite due dates (due dates for finished products' release) it is necessary to obtain a feasible schedule with the latest startup time of the technological process. With the additional constraint on the structure of a feasible schedule the exact polynomial algorithm for solving the formulated problem was stated. The theoretical and practical grounding for expedience of the introduction of additional restrictions on the structure of the desirable schedule was presented.

Введение

В статье рассматривается один класс многоэтапной сетевой задачи календарного планирования. Для произвольно заданных конечных директивных сроков (директивных сроков выпуска готовых изделий) необходимо получить допустимое расписание с максимально поздним по времени запуском технологического процесса. При дополнительном ограничении на структуру допустимого расписания излагается точный полиномиальный алгоритм решения сформулированной задачи. Приводится теоретическое и практическое обоснование целесообразности введения дополнительного ограничения на структуру искомого расписания.

Как показано в [2] первый уровень трехуровневой модели планирования и принятия решений изложенной в [1] (в терминологии [1] – третий уровень модели на котором строится полный план выполнения работ с привязкой к ресурсам – точное планирование) может быть представлен как многоэтапная сетевая задача календарного планирования. В [2] приведены пять элементов из комбинации которых конструируется многоэтапная сетевая модель календарного планирования.

Предметом исследования является следующий класс сетевых моделей:

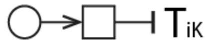
1. Сеть состоит из комбинации элементов (тип 1-4 [2]) и является ориентированным ациклочным графом.

Элементам (тип 2,3 [2]) соответствует в сети оборудование, которое в процессе непрерывной работы в произвольном порядке должны выполнить множество работ без прерывания. В произвольный момент времени может выполняться только одна работа. Элементы отличаются между собой типами связи в сети работ, выполненных оборудованием. Сеть может содержать элемент, являющийся комбинацией элементов (тип 2,3 [2]) (по структуре связей выполненных работ в сети). Элементам (тип 1,4 [2]) соответствует оборудование, выполняющее только одну работу.

2. Сама сеть представляет собой ориентированный граф с двумя типами вершин – O – обозначает работу готовую к выполнению, за ней обязательно следует вершина □ – обозначение оборудования на котором выполняется непосредственно предшествующая ему работа.

3. Все ориентированные стрелки направлены слева направо.

4. Всю сеть можно разбить на три части.

4.1 Конечная часть заканчивается элементами вида 1 [2], 

где T_{ik} – время выполнения последней операции (работы) по i-ому изделию (i-ой серии изделий). В допустимом расписании $T_{ik} \le T_i, i = \overline{1, I}$, где T_i – директивный срок выполнения i-ого изделия (i-ой серии изделий).

Примечание. Последней может быть работа, выполненная на элементе (тип 2,3 [2]).

Представляет собой произвольную комбинацию элементов 1-4, которая удовлетворяет следующему условию:

$T_i, i=\overline{1,I}$ однозначно задают для элементов (типа 2, 3[2]) директивные сроки выполнения работ, которые в произвольном порядке могут выполняться на элементах (типа 2,3) [2], у каждого из которых в конечной части сети, нет непосредственно предшествующих элементов (типа 2,3 [2]) (без учета элементов (типа 1,4 [3])).

Примечание. У элемента 1 [2], T может быть промежуточным сроком выполнения работы (не окончание выполнения изделия (группы изделий))

4.2 Промежуточная часть.

Это произвольная сетевая комбинация элементов (тип 1-4 [2]), удовлетворяющая условию: найденные ниже приведенным алгоритмом наиболее поздние моменты готовности выполнения работ на элементах (типа 2,3 [2]) однозначно задают директивные сроки выполнения работ для непосредственно предшествующих (без учета элементов (типа 1,4) [2]) элементов (типа 2,3 [2]). Непосредственное предшествование означает, что между непосредственно предшествующим элементом (типа 2,3 [2]) и данным элементом (типа 2,3 [2]), существует ориентированный путь содержащий только вершины соответствующие элементам (тип 1,4 [2]), с учетом примечания к п.4.1.

В промежуточной части сети могут находиться элементы (типа 2,3 [2]) непосредственно предшествующие более чем одному элементу (тип 2,3 [2]).

4.3 Начальная часть сети является произвольной комбинацией элементов (тип 1-4 [2] удовлетворяющая следующим условиям:

a) имеются все вершины сети вида $\boxed{1_j}$, $j=\overline{1,p}$, где цифра 1 означает что $\boxed{1_j}$ вершине не предшествует ни одна вершина второго типа (смотри пункт 2).

b) произвольному элементу (тип 2,3 [2]) непосредственно не предшествует элемент (тип 2,3 [2]).

c) рассмотрим в сети полную группу элементов (тип 2,3 [2]). Тогда найденные ниже приведенным алгоритмом наиболее поздние моменты начала выполнения работ элементов этой группы однозначно задают моменты начала выполнения работ всех элементов типа $\boxed{1_j}$, $j=\overline{1,p}$.

Примечание.

1) Сеть точно отражает реальный технологический процесс.

2) Так как выполнение работ может быть разделено во времени, в сети одно и тоже физически существующее оборудование может быть представлено различными вершинами второго типа. Так, например элементы (тип 2,3 [2]) работают непрерывно.

Реально физически существующее оборудование непрерывно выполняет работы для различных групп операций в соответствии с технологическим процессом (время их выполнения на непрерывно работающем оборудовании разнесено во времени), тогда в сети одному и тому же оборудованию соответствует несколько элементов (тип 2,3 [2]).

Приведем на рисунке 1 пример сети, принадлежащей сформулированному классу.

Необходимо решить следующую задачу: найти допустимое расписание выполнения работ ($\forall T_{ik} \le T_i, i=\overline{1,I}$), которому бы соответствовал $\max(\min_{j=\overline{1,p}} T_{1j})$ либо $\max\sum_{j=1}^{p} T_{1j}$, где T_{1j} момент запуска оборудования соответствующего вершине $\boxed{1_j}$. Для примера на рисунке 1 функционалы имеют вид: $\max(\min_{j=\overline{1,18}} T_{1j})$ либо $\max\sum_{j=1}^{18} T_{1j}$.

Для решения поставленной задачи наложим и обоснуем дополнительное ограничение на искомое допустимое расписание. Для этого рассмотрим свойства следующей одноэтапной задачи теории расписаний [3], являющейся моделью элементов (тип 2,3 [2]).

Постановка задачи

Задано множество независимых заданий $J = \{ j_1, j_2, ..., j_n\}$, каждое из которых состоит из одной операции. Для каждого задания j известны длительность выполнения p_j и директивный срок выполнения d_j. Прерывания не допускаются. Задания поступают в систему одновременно. Необходимо:

1) найти максимально поздний момент r начала выполнения заданий в допустимом расписании;

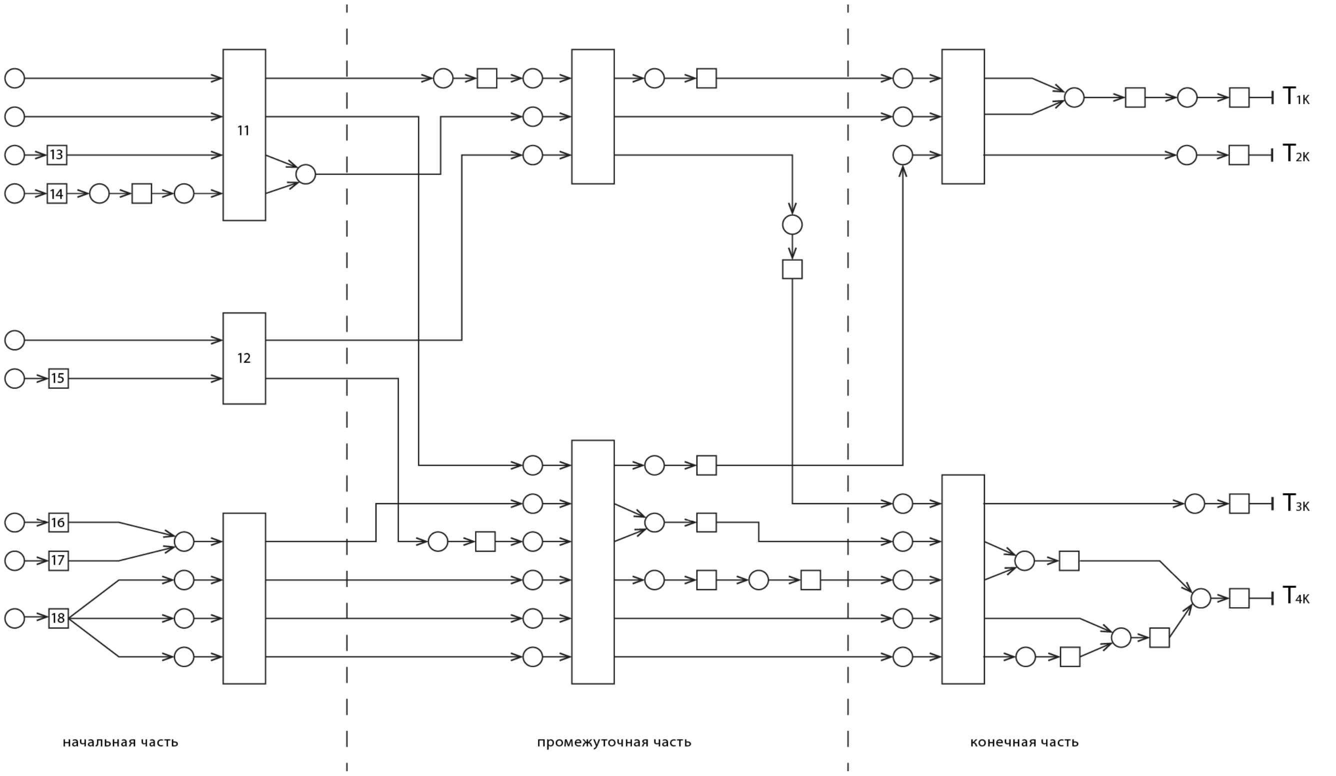


Рис. 1. Пример сети

2) минимизировать суммарное опережение выполнения заданий относительно директивного срока при условии допустимости расписания.

Определение 1. Если для заданного момента выполнения заданий объективно существует расписание, в котором удовлетворены все директивные сроки, то такое расписание называется допустимым.

Теорема 1. [3] Пусть существует произвольная допустимая последовательность выполнения заданий одним прибором $\sigma_1 = \{j_{[1]}, j_{[2]}, ..., j_{[n]}\}$, где $j_{[k]}$ обозначает задание на позиции k в текущей последовательности. Последовательность на этом множестве заданий, упорядоченная по неубыванию значений директивных сроков, также допустима.

Следствие. Если расписание, упорядоченное по неубыванию директивных сроков, является недопустимым, то допустимого расписания не существует.

Алгоритм А [3]

4) Строим, в соответствии с теоремой 1, расписание J = $\{1, 2, \ldots n\}$, в котором задания упорядочены по неубыванию директивных сроков.

5) Определяем момент запуска выполнения заданий r, при котором в расписании J хотя бы одно задание j_k будет запаздывающим:

$$\exists k : d_k - r < \sum_{i=1}^{k} p_i .$$

6) Для момента запуска r находим задание j_l с максимальным запаздыванием относительного директивного срока и определеляем новый момент начала выполнения заданий $r_{\max} = r - \left(C_l - d_l\right)$, где $C_l - d_l$ – запаздывание по заданию j_l.

Теорема 4.[3] Допустимое расписание, построенное по неубыванию директивных сроков, является оптимальным по критерию минимизации суммарного опережения при выполнении условий: $d_1 \le d_2 \le \ldots \le d_n$, $p_1 \ge p_2 \ge \ldots \ge p_n$. Суммарное опережение минимально при моменте запуска $r_{\max}$.

Примечание. Если в допустимом расписании $\sigma = \{1, 2, ..., n\}$ с максимальным моментом начала выполнения заданий для произвольного C_k существуют задания $i \in \overline{1, k-1}$, для которых $C_k \le d_i$, то при упорядочении этих заданий по невозрастанию их длительностей получаем допустимое расписание с максимальным моментом начала выполнения заданий, в котором уменьшается суммарное опережение (в предположении, что в σ существуют как минимум две такие работы различной длительности, не упорядоченные по невозрастанию их значений).

$$\sum_{i=1}^{n-1}\left\lfloor\left(d_i - r_{\max} - \sum_{j=1}^{i} p_j\right)\Big/\min_{j=\overline{i+1,n}} p_j\right\rfloor \Delta_i, \quad \Delta_i = \max\left(0, \max_{j=\overline{i+1,n}} p_j - p_i\right), \qquad (1)$$

Теорема 5. [3] В общем случае верхняя оценка отклонения суммарного опережения допустимого расписания, полученного упорядочением по неубыванию директивных сроков, от оптимального расписания для того же момента запуска по критерию минимизации суммарного опережения, определяется выражением:

где $\lfloor a \rfloor$ обозначает ближайшее снизу целое от a.

С учетом выше приведенных свойств одноэтапной задачи календарного планирования (модели элементов (тип 2,3 [2])) на допустимое расписание многоэтапной сетевой задачи календарного планирования накладывается следующее дополнительное ограничение (Д ограничение):

1. Расписание выполнения работ на элементах (тип 2,3 [2]) всегда является допустимым при известных (произвольно назначенных, найденных алгоритмом) директивных сроках выполнения работ.
2. Моментом запуска непрерывной работы элемента (тип 2,3 [2]) является $r_{\max}$, определяемый Алгоритмом А.
3. Расписание работы элементов (тип 2,3 [2]) определяются расписанием (1,2,…,n) (Теорема 1) с учетом упорядочения работ (смотри примечание к теореме 4, которое реализуется последовательно от меньшего к большему k, для $\forall C_k, k = \overline{2,n}$).
4. Самые поздние сроки моментов готовности работ на элементе (тип 2, 3 [2]), определяемые расписанием их вычисления, однозначно задают директивные сроки выполнения работ на элементах (тип 2, 3 [2]) непосредственно ему предшествующих.

Алгоритм построения субоптимального полиномиального алгоритма решения многоэтапной сетевой задачи календарного планирования

1. По моментам времени $T_i, i = \overline{1, I}$, I – количество изделий (серий изделий) в силу свойств сети однозначно определяются все директивные сроки выполнения работ всех элементов (типа 2,3 [2]) конечной части сети.
2. Расписание работ всех элементов (типа 2,3 [2]) конечной части сети находятся в соответствии с Д ограничением.
3. По построенным расписаниям находятся все самые поздние моменты готовности на выполнения работ непосредственно предшествующие элементам (типа 2,3 [2]) конечной части сети.
4. В силу свойств сети по найденным моментам готовности однозначно определяются все директивные сроки выполнения работ элементов (типа 2,3 [2]) непосредственно предшествующих (связанных хотя бы одним направленным путем не содержащих элементов (типа 2,3 [2])).
5. Расписание элементов (типа 2,3 [2]) у которых определены все директивные сроки определяется Д ограничением.
6. Пункты 4,5 повторяются (не более чем общее количество элементов (типа 2,3 [2]) промежуточной и начальной части сети) до получения всех самых поздних моментов готовности на выполнение работ элементов (типа 2,3 [2]) начальной части сети.
7. В силу свойств сети по этим моментам времени однозначно находятся моменты времени выполнения всех начальных работ. Конец.

Допустимое расписание построено.

Свойство полученного расписания

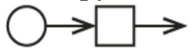
1. Алгоритм построения допустимого расписания является полиномиальным.
2. В силу логики построения алгоритма с учетом Д ограничения каждый из полученных моментов начала выполнения работ элементами $\boxed{1_j}$, $j = \overline{1,k}$ k – количество начальных вершин сети второго типа, являются максимально поздними , что позволяет определить полученное допустимое расписание как субоптимальное по обоим критериям $\max(\min_{j=\overline{1,p}} T_{1j})$, $\max \sum_{j=1}^{p} T_{1j}$, (оптимальное с учетом выполнения Д ограничения).

Примечание. Свойство 2 гарантированно имеет место, когда для всех элементов (типа 2,3 [2]) сетевой модели все работы назначенные на выполнение одним элементом (типа 2,3 [2]) имеют различные длительности. Действительно, в этом случае Д ограничение приводит к единственному расписанию для каждого элемента (типа 2,3 [2]).

3. Расписания всех элементов (типа 2,3 [2]) начальной и промежуточной части с учетом назначенных алгоритмов директивных сроков, обладают свойствами вытекающими из теорем (1-5) [3].

4. Свойства полученного субоптимального расписания относительно конечных директивных сроков $T_i, i = \overline{1, I}$ отвечает всем свойствам, вытекающих из теорем (1-5) [3].

Действительно, суммарное опережение построенного расписания является суммой суммарных опережений расписаний всех элементов (типа 2,3[2]) конечной части сети, директивные сроки которых однозначно определяются по $T_i, i = \overline{1, I}$, а сами расписания построены в соответствии с Д ограничением.

Примечание. Эффективность предложенного расписания полностью определяется адекватностью предложенной сетевой модели реальному технологическому процессу и физически существующему оборудованию. Если в сети элементы типа  однозначно задаются технологическим процессом и физически существующим оборудованием, то анализ полученного расписания может показать, что некоторые элементы (тип 2,3[2]) во время непрерывной работы фактически выполняют разнесенные во времени различные группы работ. Тогда необходимо изменить модель сети, введя дополнительные виртуальные элементы (типа 2,3[2]), соответствующие этому оборудованию и заново решить задачу.

Выводы

Сформулирован класс многоэтапных сетевых задач календарного планирования, для которого приведен и обоснован субоптимальный алгоритм, исследованы теоретические свойства полученного решения.

Список литературы

1. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография.– К.: Наукова думка. – 2010. – 573 с.
2. Павлов А.А., Мисюра Е.Б., Халус Е.А., Сперкач М.О., Аракелян Г.А. Результирующая формализация первого уровня трехуровневой модели оперативного планирования и принятия решений по критерию минимизации суммарного опережения директивных сроков // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. К.: «ВЕК+», 2012. « No.56.» 2 С. (подано до друку)
3. Павлов А.А., Мисюра Е.Б., Халус Е.А. Полиномиальный алгоритм получения допустимого расписания с максимально поздним моментом начала выполнения одним прибором независимых заданий произвольной длительности с разными директивными сроками, при котором все задания остаются незапаздывающими // Інформаційні технології як інноваційний шлях розвитку України у XXI столітті: Матеріали І Міжнародній науково-практичній конференції молодих науковців 06-08 грудня 2012 р. – Ужгород: Закарпатський державний університет, 2012. – 7 С.